Claims

What is claimed is:

1. A system for communicating over a protocol, comprising:

a content associated with a Uniform Resource Identifier (URI) to be exposed; and

a class factory comprising a plurality of identifiers and associated registered listener object creators, at least one of the listener object creators adapted to create at least one listener object that facilitates exposure of the URI.

2. The system of claim 1, comprising:

a reading component that reads a first data from an accessing application, the first data having at least one of, a format specific to the protocol and one or more headers and/or footers specific to the protocol, when the first data is read from the accessing application, the first data requesting access to a second data;

an exposing component that exposes the second data, the second data being associated with a URI; and

a writing component that writes a third data to the accessing application, the third data having at least one of, a format specific to the protocol and one or more headers and/or footers specific to the protocol, when the third data is written to the accessing application, the third data being derived, at least in part, from the second data.

3. The system of claim 1, comprising:

an exposing component that exposes a resource to access by one or more accessing applications.

4. The system of claim 3, wherein the resource is at least one of, a service, an application and a content source, and the resource is accessible over a network.

5. The system of claim 4, wherein the protocol is at least one of a Hypertext Transfer Protocol (HTTP), a File Transfer Protocol (FTP) and a Simple Mail Transport Protocol (SMTP).

6. The system of claim 5, wherein the plurality of identifiers comprise one or more URIs.

7. The system of claim 5, wherein the plurality of identifiers comprise one or more prefixes associated with one or more URIs.

8. The system of claim 1, wherein at least one of the listener object creators instantiates at least one listener object, with at least one of the listener object creators being software in execution.

9. The system of claim 8, wherein the at least one of the listener object creators registers one or more implemented creating methods with the class factory, the creating methods being defined in an abstract base class and implemented by the at least one listener object.

10. The system of claim 9, wherein the at least one listener object inherits from one or more abstract base classes.

11. The system of claim 2, wherein the at least one listener object is adapted to listen for the first data from the accessing application.

12. The system of claim 11, wherein the at least one listener object makes the first data received from the accessing application available as a byte stream to a server program.

13. The system of claim 12, wherein the at least one listener object removes at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol from the first data.

14. The system of claim 13, further comprising:

one or more protocol objects adapted to write the third data to the accessing application.

15. The system of claim 14, wherein the at least one protocol object accepts a byte stream to write as the third data to the accessing application.

16. The system of claim 15, wherein the at least one protocol object adds at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol to the third data.

17. The system of claim 1, comprising:

an adding component adapted to add one or more identifiers to a list of registered identifiers, and further adapted to add one or more listener object creating methods to a list of registered listener object creating methods.

18. The system of claim 1, wherein the at least one listener object and one or more answering objects reside on a single server machine, the at least one listener object and the one or more answering objects operable to expose one or more applications, services and/or content sources to one or more accessing applications.

19. A method for allowing a server program to listen for and process requests received over one of a plurality of protocols, comprising:

registering one or more protocol handlers operable to create a listener object;

creating an instance of a listener object from a source of registered protocol handlers;

returning the listener object to the application, the listener object being an implementation of a base class;

exposing one or more URIs through functionality provided by the listener object; and

listening for a request from an accessing application through functionality provided by the listener object.

20. The method of claim 19, further comprising:

registering one or more protocol handlers operable to create an answering object;

creating an instance of an answering object from a source of registered protocol handlers;

returning the answering object to the application, the answering object being an implementation of a base class; and

writing a data responsive to the request to the accessing application *via* the answering object.

21. The method of claim 19, further comprising:

a computer program generating a request to expose a resource, wherein the request to expose contains a Universal Resource Identifier (URI) to identify the resource.

22. The method of claim 21, wherein the resource is at least one of, an application, a service, and a content source.

23. The method of claim 19, wherein creating an instance of a listener object from a source of registered protocol handlers comprises:

selectively determining one or more listener object creators operable to create the listener object based, at least in part, on a portion of a URI, where the listener object creator implements one or more creator methods defined in an abstract creator base class; and

invoking at least one of the one or more listener object creators to create the listener object.

24. The method of claim 19, wherein using the base class API to communicate through the listener object comprises:

implementing one or more methods defined in the base class API in a derived class; and

employing the one or more implemented methods.

25. The method of claim 24, wherein at least one of the one or more methods can be employed to read a first data from an accessing application, the first data having at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol when it is read from the accessing application, the first data being provided to a server program as a byte stream.

26. The method of claim 25, wherein at least one of the one or more methods can be employed to write a third data to the accessing application, the third data having at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol when it is written to the accessing application, the third data being provided to the one or more methods as a byte stream, the third data being responsive to the first data, and the third data being derived from a second data, the second data being associated with the one or more exposed URIs.

27. A computer readable medium having computer executable instructions operable to perform the method of claim 19.

28. A computer readable having computer executable instructions operable to perform the method of claim 26.

29. A data packet adapted to be transmitted between two or more computer processes, the data packet comprising:

information operable to facilitate selecting a listener object creator.

30. A data packet adapted to be transmitted between two or more computer processes, the data packet comprising:

a byte stream produced by a listener object, the byte stream having at least one of a format specific to a protocol and one or more headers and/or footers specific to the protocol removed from a first data read from an accessing application.

31. The data packet of claim 30, further comprising:

a second data having at least one of a format specific to the protocol and one or more headers and/or footers specific to the protocol added to a byte stream provided by a server program, the second data responsive to the first data.

32. A system for simplifying a server program exposing a resource over a protocol, comprising:

storing means for storing a data related to resolving a Uniform Resource Identifier;

registering means for registering a listener object creator;

creating means for creating a listener object;

determining means for selectively determining means for creating a listener object;

accessing means for accessing a method in the listener object, which method implements a method defined in a network object base class; and

communicating means for communicating with an accessing application, wherein the communicating means employ one or more methods in the listener object.